Benefits of Natural Language Techniques in Ontology Evaluation : the OOPS! Case

Maria Carmen Suarez-Figueroa, Mouna Kamel, María Poveda-Villalon

Benefits of Natural Language Techniques in Ontology Evaluation: the OOPS! Case

Mari Carmen Suárez-Figueroa
Ontology Engineering Group. Universidad Politécnica de Madrid
mcsuarez@fi.upm.es

Mouna Kamel
Institut de Recherche en Informatique de Toulouse (IRIT) - CNRS
Mouna.Kamel@irit.fr

María Poveda-Villalón
Ontology Engineering Group. Universidad Politécnica de Madrid
mpoveda@fi.upm.es

Abstract

Natural language techniques play an important role in Ontology Engineering. Developing ontologies in a manual fashion is a complex and time consuming process, which implies the participation of domain experts and ontology engineers to build and evaluate them. Natural language techniques traditionally help to (semi)-automatically build ontologies and to populate them. However, the general trends for evaluating ontologies are mainly expert reviewing, evaluating quality dimensions and criteria, and evaluating against existing ontologies and set of common errors. That is, the use of natural language techniques in ontology evaluation is not widely spread. Thus, in this paper we aim at the use of natural language techniques during the ontology evaluation process. In particular, we propose a first attempt towards a language-based enhancement of the pitfall detection process within the ontology evaluation tool OOPS!.

1 Introduction

Developing ontologies manually is a complex and time consuming process, which involves both ontology engineers and domain experts. Natural language (NL) techniques have been traditionally used for extracting knowledge from texts to build semantic resources. In fact knowledge acquisition from text plays an important role in Ontology Engineering. It is divided into several steps, according to the "ontology learning layer-cake" (Cimiano, 2006): (a) identifying and extracting terms, (b) eliciting concepts and relations linking concepts from these terms, (c) organizing concepts and relations into hierarchies, and (d) identifying axioms.

During the ontology building, a wide range of difficulties and handicaps can appear. These situations may have as consequence the inclusion of anomalies in the ontology. Thus, the ontology evaluation process plays a key role in ontology engineering developments. Currently, the general trends in ontology evaluation involve different approaches (e.g., the comparison of the ontology to a "gold standard" or the detection of common errors in the ontology). However, what seems to be less present in the ontology evaluation field is the intensive use of NL techniques. For example, some structural or naming errors in the ontology may be automatically pointed out with a linguistic analysis of concept labels. Thus, our intention in this paper is to aim at the use of NL techniques during the ontology evaluation process. In particular, we propose a first attempt of improving the pitfall detection methods implemented within OOPS! by means of NL techniques.

The remainder of this paper is structured as follows: Section 2 summarizes different NL techniques used in Ontology Engineering. Section 3 presents the relation between ontology evaluation and NL-based techniques. Section 4 briefly describes OOPS!. In Section 5 our proposal towards a language-based enhancement of the pitfall detection process within OOPS! is presented. Finally, Section 6 outlines some conclusions and future steps.

2 Natural Language Techniques in Ontology Engineering

NL techniques traditionally help on the (semi)-automatic building of ontologies and on the population of ontologies with instances.

Most of the approaches for building ontologies from text, known as ontology learning methods, usually implement lexico-syntactic patterns (Hearts, 1992; Montiel-Ponsoda and Aguado de Cea, 2010), clustering methods or machine learning algorithms (essentially unsupervised) (Poelmans et al., 2010), to exploit various linguistic clues. Some platforms exist and implement one or a combination of these methods using different NLP tools (term or relation extractors, parsers, etc.). Examples are Text2Onto (Cimiano and Völker, 2005), which discovers concepts and hyperonimic relations between concepts, thanks to lexico-syntactic patterns and associative rules automatically learned from examples and OntoLearn (Velardi et al., 2005), which uses Wordnet (Fellbaum, 1998) for identifying lexical relations.

Regarding the population of ontologies, tools like TEXCOMON (Zouaq and Nkambou, 2008) uses linguistic patterns for instance identification, using named entity recognition techniques.

Linguistic approaches have been also applied to ontology matching where Euzenat and Shvaiko (2007) distinguish between language-based methods and methods which are based on linguistic resources, whereas the more general class of terminological approaches also includes string-based methods. We can mention the work by Ritze et. al (2010) that shows how complex matching can benefit from NL techniques.

3 Ontology Evaluation and Natural Language Techniques

Ontology evaluation process, which checks the technical quality of an ontology against a frame of reference (Suárez-Figueroa, 2010), plays a key role in ontology engineering projects.

To help developers during the ontology evaluation process, there are different approaches (Sabou and Fernandez, 2012; Poveda-Villalón et al., 2012): (a) comparison of the ontology to a "gold standard", (b) detection of common errors from catalogues in the ontology, (c) use of dimensions and criteria for describing the quality and goodness of the ontology, (d) use of the ontology in an application and evaluation of the results, (e) comparison of the ontology with a source of data about the domain to be covered, and (f) evaluation by experts who check the ontology against the requirements.

In addition, ontology evaluation can be supported by NL techniques in several ways (Gangemi et. al, 2005):

- When the ontology directly supports information retrieval or text mining applications and thus concerns objects mentioned in texts.
- When a corpus of documents is available, NLP can be used to identify mentions of instances (i.e. occurrences in text) of classes and relations which are mentioned in the text. A corpus-based evaluation of the ontology can reveal important properties of the ontology that might not be discovered otherwise.
- When (semi)-automatic population of the ontology is performed, NLP can help in the identification of new senses of already known instances, for example because the instance is polysemous and/or ambiguous (e.g., "Washington" is a person and a location).

However, ontology evaluation approaches could take more advantage of NL techniques. In this sense, we propose here a first attempt towards a NL-based upgrade of OOPS!.

4 OOPS!: OntOlogy Pitfall Scanner!

OOPS![1] (Poveda-Villalón et al., 2012) is a web-based tool, independent of any ontology development environment, for detecting potential pitfalls that could lead to modelling errors. Currently, OOPS! provides mechanisms to automatically detect as many pitfalls as possible, thus it helps developers in the diagnosis activity, which is part of the ontology validation process.

OOPS! takes as input an ontology to be evaluated and a pitfall catalogue in order to produce a list of evaluation results. The current version of the catalogue[2] consists on 35 pitfalls. Some examples are creating synonyms as classes, defining wrong inverse relationships, missing annotations, missing domain or range in properties, or defining wrong equivalent classes. Up to now, OOPS! detects semi-automatically a subset of 21 pitfalls related to the following dimensions: human understanding, logical consistency, modelling issues, ontology language specification and real world representation.

[1] http://oeg-upm.net/oops/
[2] http://www.oeg-upm.net/oops/catalogue.jsp

5 Towards a Language-based Enhancement of OOPS!

In this section we propose a first attempt towards a language-based enhancement of the pitfall detection process within the ontology evaluation tool OOPS!. To do this, we have reviewed the current catalogue of pitfalls in order to determine (a) which pitfalls, already implemented, could be detected in a better way by means of applying linguistic techniques and (b) which ones, not detected yet by OOPS!, could be implemented based on linguistic aspects.

Regarding the proposals for enhancing pitfalls already detected by OOPS!, we can mention the following ones:

- *P2. Creating synonyms as classes*: several classes whose identifiers are synonyms are created and defined as equivalent. Its detection could be improved by using linguistic resources such as WordNet and EuroWordNet, particularly by looking for the synonymy information of the class name.

- *P3. Creating the relationship "is" instead of using "rdfs:subClassOf", "rdf:type" or "owl:sameAs"*: the "is" relationship is created in the ontology instead of using OWL primitives for representing the subclass relationship ("subclassOf"), the membership to a class ("instanceOf"), or the equality between instances ("sameAs"). The detection could be enriched by creating specific language-dependent lexico-syntactic patterns to discover the use of 'is' and by using named entity recognition tools for characterizing the "instanceOf" relation.

- *P5. Defining wrong inverse relationships*: two relationships are defined as inverse relations when they are not necessarily. As first attempt, the implementation of this pitfall could be improved by creating specific lexico-syntactic patterns for direct/inverse relationship name structure.

- *P7. Merging different concepts in the same class*: a class is created whose identifier is referring to two or more different concepts (e.g., "StyleAndPeriod", or "ProductOrService"). As first attempt, its detection could be enhanced by creating specific language-dependent lexico-syntactic patterns and regular expressions to discover the use of 'and' or 'or' in the concept name.

- *P12. Missing equivalent properties*: when an ontology is imported into another, developers normally miss the definition of equivalent properties in those cases of duplicated relations and attributes (e.g., "hasMember" and "has-Member" in two different ontologies). The detection could be enriched by (a) using linguistic resources such as WordNet and EuroWordNet, specifically by looking for the synonymy information of the property name and (b) creating specific language-dependent lexico-syntactic patterns.

- *P13. Missing inverse relationships*: this pitfall appears when a relationship (except for the symmetric ones) has not an inverse relationship defined within the ontology. As first attempt, its implementation could be improved by creating specific lexico-syntactic patterns for direct/inverse relationship name structure (e.g., isSoldIn-sells; hasAuthor-isAuthorOf; hasParent-isParentOf).

- *P21. Using a miscellaneous class*: to create in a hierarchy a class that contains the instances that do not belong to the sibling classes instead of classifying such instances as instances of the class in the upper level of the hierarchy. This class is normally named "Other" or "Miscellaneous". As first attempt, its detection could be improved by creating a set of lexico-syntactic patterns that represent different ways of naming concepts that are usually miscellaneous entities.

With respect to those pitfalls not detected yet by OOPS!, we can propose the following ideas for their implementation based on NL aspects:

- *P1. Creating polysemous elements*: an ontology element whose name has different meanings is included in the ontology to represent more than one conceptual idea. As first approach, its detection could be implemented by (a) using linguistic resources such as WordNet and EuroWordNet, specifically by analyzing the different synsets in which the element name appears and (b) by analysing labels of neighbourhood concepts for disambiguation.

- *P9. Missing basic information*: information that is required and/or useful is not included in the ontology. As first approach and in certain situations, this pitfall could be implemented by using linguistic resources such as WordNet and EuroWordNet, specifically by analyzing the antonym information of the relationships name.

- *P30. Missing equivalent classes*: when an ontology is imported into another, classes with the same conceptual meaning that are duplicated in both ontologies should be defined as equivalent classes. As first step, this pitfall could be detected by using linguistic resources such as WordNet and EuroWordnet, specifically by looking for the synonymy information of the class name.

• *P31. Defining wrong equivalent classes*: two classes are defined as equivalent when they are not necessarily. As first step, this pitfall could be implemented by using linguistic resources such as WordNet and EuroWordNet, specifically by looking for the hyperonym information of the class name.

6 Conclusions and Future Work

In this paper, we have presented the first efforts towards a NL-based enhancement of the pitfall detection process within the ontology evaluation tool OOPS!. We have reviewed the 35 pitfalls in the OOPS! catalogue and analyzed which pitfall detections could be linguistically improved and which pitfalls could be implemented based on NL as first attempt. In summary, we have proposed the improvement of 7 pitfall detection processes and the automation of 4 pitfalls not detected yet by OOPS!. Thus, we have planned to enhance OOPS! with the NL techniques presented in this paper.

7 ACKNOWLEDGMENTS

This work has been supported by the Spanish project BabelData (TIN2010-17550).

References

Philipp Cimiano. 2006. Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer, November.

Philipp Cimiano and Johanna Völker. 2005. Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery. in A. Montoyo, R. Munoz, E. Metais (eds), Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB), vol. 3513 of LNCS, Springer, Heidelberg, p. 227-238.

Jérôme Euzenat and Pavel Shvaiko. 2007. Ontology Matching. Springer.

Christiane Fellbaum (ed.). 1998. Wordnet. An Electronic Lexical Database. The MIT Press, Cambridge Massacuhsets.

Aldo Gangemi, Carola Catenacci, Massimiliano Ciaramita, and Jens Lehmann. 2005. Ontology evaluation and validation: An integrated formal model for the quality diagnostic task. Technical report, Laboratory of Applied Ontologies. CNR, Rome, Italy.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In Procs. of the 14th International Conference on Computational Linguistics COLING1992, p. 539–545.

Elena Montiel-Ponsoda and Guadalupe Aguado de Cea. 2010. Using natural language patterns for the development of ontologies. In V. Bhatia, P. Sánchez-Hernandez & P. Pérez Paredes, Eds., Researching specialized languages, p. 211–230. John Benjamins Pub.

Jonas Poelmans, Paul Elzinga, Stijn Viaene, and Guido Dedene. 2010. Formal concept analysis in knowledge discovery: a survey. In Proceedings of the 18th international conference on Conceptual structures: from information to intelligence, ICCS'10, p. 139–153, Berlin, Heidelberg: Springer-Verlag.

María Poveda-Villalón, Mari Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. 2012. Validating ontologies with OOPS!, in Knowledge Engineering and Knowledge Management, Vol. 7603, Lecture Notes in Computer Science (Springer-Verlag, Berlin, 2012), pp. 267–281.

Dominique Ritze, Johanna Völker, Christian Meilicke, and Ondřej Šváb-Zamazal. 2010. Linguistic analysis for complex ontology matching. Proceedings of the 5th International Workshop on Ontology Matching (OM-2010), Shanghai, China.

Marta Sabou and Miriam Fernandez. 2012. Ontology (Network) Evaluation. Ontology Engineering in a Networked World. Suárez-Figueroa, M.C., Gómez-Pérez, A., Motta, E., Gangemi, A. (Editors). Pp. 193-212, Springer. ISBN 978-3-642-24793-4.

Mari Carmen Suárez-Figueroa. 2010. PhD Thesis: NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse. Spain. Universidad Politécnica de Madrid.

Paola Velardi, Paolo Fabriani and Michele Lissikof. 2005. Using text processing techniques to automatically enrich a domain ontology. In Proceedings of ACM- FOIS., Maine, publication of ACM, 270 – 284.

Amal Zouaq and Roger Nkambou. 2008. Building Domain Ontologies from Text for Educational Purpos-es. IEEE Transactions on Learning Technologies 1(1), pp. 49 –62.